
Lab4 HTML-PHP Lab

Table of Contents

Introduction.....	2
<i>Summary for Lab4, and the next Lab (Lab5).....</i>	<i>2</i>
<i>Goal for this Lab.....</i>	<i>2</i>
<i>Lab Report:.....</i>	<i>2</i>
How to screenshot the active application.....	2
<i>Required Software</i>	<i>2</i>
<i>Resource</i>	<i>2</i>
Part1: Configure your personal web page	4
1. Mount the multidrive on your personal computer	4
2. Setup your personal web page https://classdb.it.mtu.edu/~YOUR_USER_NAME	4
3. Create a html helloworld.html under classdb.....	4
Part2: HTML	6
4. Create file login.html	6
5. Create file main.html	6
6. (Fall 2023 - optional) Create quiz.html	6
7. Create question.html	6
Part3: Server Side Basic PHP Programing	7
8. How to debug?	7
9. Implement the login function to authenticate user	8
10. How to force users to login first when they try to visit main.php?	9
11. How to implement the logout function?.....	10
12. (Optional in Fall 2023) Implement the submit function in quiz.html.....	11
13. Implement the submit function in question.html.....	12

Introduction

Summary for Lab4, and the next Lab (Lab5)

During these labs, we will implement a simple web database application. We separate the implementation of the mini bank system into different modules:

1. Authentication module (login.php)
2. Main control module (main.php)
3. Business module (bankoperation.php)
4. Database Operation module (db.php).

The first two parts will be completed in this lab, while the last two parts will be covered in the next lab.

Goal for this Lab

In this lab, you will engage in the following exercises:

1. Understand the required minimum permissions of directories and files for a web server
2. Client-side programming:
 - a. Create html files with forms of different input types
 - b. Understand the Get and Post method
3. Server-side programming:
 - a. Program with PHP on the server end
 - b. Use form data (`$_POST`)
 - c. Use sessions to keep information across multiple pages (`session_start()`, `$_SESSION`)
4. We will also practice work with some html which should help you to do your course project.

Lab Report:

You don't need to write report for each step. At the end of the lab session, include the following urls, screenshots and source code, using the provided template file.

How to screenshot the active application

Windows: Alt+PrtSc The screenshot will be save to the clipboard.
or Win + Shift + S keys for specific areas.

Mac: Command+Shift+4, then spacebar. then your mouse cursor will turn into a camera icon. Move the camera icon to the active window and click it. After that, the screenshot of the window will be immediately saved to the desktop in PNG format.

Required Software

- BIG_IP_Edge Client (VPN client)
- Multi Drive (Mount the M:drive /local/my_web_files/your_user_name) on to your local computer)
- Visual Studio Code (Edit html and php files)

Resource

HTML:

Cookies:

https://en.wikipedia.org/wiki/HTTP_cookie

https://www.w3schools.com/php/php_cookies.asp

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>

PHP

General Resource: <https://www.w3schools.com/php/>

Sessions in PHP: https://www.w3schools.com/php/php_sessions.asp

`parse_ini_file()`: <https://www.php.net/manual/en/function.parse-ini-file.php>

Part1: Configure your personal web page

1. Mount the multidrive on your personal computer

Please follow the instruction to mount the drive.

- [Mounting H: and M: Drives on Windows Machines.](#)
- [Connecting to Multidrive on a Linux machine](#)
- [Connecting to Multidrive on a Mac](#)

2. Setup your personal web page https://classdb.it.mtu.edu/~YOUR_USER_NAME

2.1 On your personal computer, under the drive M:\my_web_files, follow the instructions in:

<https://classdb.it.mtu.edu/webpage-setup.php>

2.1.1 create dir classdb,

2.1.2 create configuration file .htaccess inside the classdb directory. The file name is .htaccess (the first letter is dot), which is called a hidden file.

2.1.3 Create a simple file index.php inside the classdb directory.

2.1.4 Open the page in your browser: <https://classdb.it.mtu.edu/~yourname/index.php>

If you get "Forbidden You don't have permission to access this resource", see 2.2 about how to check and fix permission. After you fix the permission, try to visit the page again until it is working.

2.2 Verify/Change permissions

In order for the web server to access the files, we need to give Apache user appropriate access permission.

- Execute permission to all directory
- Read permission to .htaccess, all html, and php files.
- This needs to be checked/modified on the lab computers.

2.2.1 Remote login to wopr.csl.mtu.edu via ssh

2.2.2 Verify/Set the permission for dir classdb to be "executable" for "Others"

```
cd /local/my_web_files/YOUR_USRNAME
```

```
ls -ld classdb
```

```
chmod o+x classdb
```

```
ls -l .htaccess
```

```
chmod o+r .htaccess
```

```
ls -l index.php
```

```
chmod o+r index.php
```

3. Create a html helloworld.html under classdb

3.1 Create the file with the content below

```
<html>
<body>
  <h1>Chapter 1</h1>
  <p>hello. My name is Joe. I work in a button factory....</p>
</body>
</html>
```

3.2 Check and change permission to be world readable

```
ls -l helloworld.html
chmod o+r helloworld.html
```

3.3 Access it at classdb.it.mtu.edu/~yourusername/helloworld.html

Part2: HTML

4. Create file login.html

username:

password:

5. Create file main.html

Welcome to our online minibank!

We can help you to tranfer the money or display your accounts.

What would you like to do? Please click one of the buttons

6. (Fall 2023 - optional) Create quiz.html

Show the following three questions and a submit button. If you like, you are encouraged to use CSS to style the page. See the tips about CSS at the end of the document.

Q1: The pace of this course

☒ A: is too slow

☐ B: is just right

☐ C: is too fast

☐ D: I don't know

Q2: The feedback from homework assignment grading

☐ A: too few

☒ B: sufficient

☐ C: I don't know

Q3: Any thing you like about the teaching of this course?

7. Create question.html

Show the following all questions and a remove button for each question

Q1: The pace of this course

Q2: The feedback from homework assignment grading

Q3: Anything you like about the teaching of this course?

Part3: Server Side Basic PHP Programing

8. How to debug?

The basic debugging method – print the data

- a) Print the data of global variable such as \$_POST, \$_SESSION
- \$_POST: data from the last submitted form
 - \$_SESSION: data in the current session

```
<?php
echo "<pre>";
print_r($_POST);
print_r($_SESSION);
echo "</pre>";
?>
```

- b) Use echo statement to print some debugging information

Check the page source

Since the PHP code will be executed on the server side, you will not see the PHP code on the client side, only the output of the PHP code. Many time you need to see the page source on the client side to trouble shoot the bugs in PHP source code.

View HTML Source Code:

Right-click in an HTML page and select "View Page Source" (in Chrome) or "View Source" (in Edge), or similar in other browsers. This will open a window containing the HTML source code of the page.

Inspect an HTML Element:

Right-click on an element (or a blank area), and choose "Inspect" or "Inspect Element" to see what elements are made up of (you will see both the HTML and the CSS). You can also edit the HTML or CSS on-the-fly in the Elements or Styles panel that opens.

9. Implement the login function to authenticate user

9.1 Rename login.html to login.php because we will add PHP code to it.

9.2 Modify the <form> tag to define the submission action is "login.php" with method \$_POST.
Please note, the login form and action are all in the same file login.php.

```
<form method=post action=login.php>
```

Since the action is the same file, you may also use the php global variable
`action="<?php echo htmlentities($_SERVER['PHP_SELF']); ?>"`

If you don't define action, it will default to the itself.

9.3 Add PHP code to check user name and passwd, and redirect to a new page

Here is the logic:

```
IF "login" button was clicked previously when the current page is requested THEN
  Check if username and password that were entered from previously are correct
  IF correct THEN
    Go to page "main.html".
  ELSE
    Display the error message in red

Displaying the login form.
```

Please hardcoded the correct username and password for now. We will modify the code to compare with the username and password in the database tables in the next lab.

Note: use isset() function to see if a value has been defined for a variable.

Example: `isset($_POST["login_button_name"])`

<https://www.php.net/manual/en/function.isset.php>

You may use the code here as a reference:

```
<?php
// user clicked the login button */
if ( isset($_POST["login"]) ) {
    //check the username and passwd, if correct, redirect to main.php page
    if ($_POST["username"]=="Mary" && $_POST["password"]=="Hello") {
        header("LOCATION:main.html");
        return;
    }else {
        echo '<p style="color:red">incorrect username and password</p>';
    }
}

?>
```


10. How to force users to login first when they try to visit main.php?

The answer is to use Session! Add all related pages to the Session!

10.1 First rename main.html to main.php as we will add PHP code to it.

10.2 At the beginning of the login.php, start a new session/join an existing session

```
<?php
session_start();
?>
```

10.3 In login.php, set session variable \$_SESSION["username"] to remember the logged in username. Find the right place to add this line:

```
$_SESSION["username"]=$_POST["username"];
```

10.4 In main.php, join the session as you did for login.php.

10.5 In main.php, check if user is logged in by checking if \$_SESSION["username"] is set.

- If user is logged in, then display a welcome message and a log out button on the right top corner.
- If not, redirect to login.php page using header() function.

See <https://www.php.net/manual/en/function.header.php>

Reference the code below:

```
<?php
    if (!isset($_SESSION["username"])) {
        header("LOCATION:login.php");
    }else {
        echo '<p align="right"> Welcome '. $_SESSION["username"].'</p>';
    }
?>

    <form >
    <p align="right">
    <input type="submit" value="logout" name="logout">
    </p>
    </form> ;

<?php
}
?>
```

Please note, for debugging purpose, you may want to print global variable \$_SESSION and \$_POST after joining the session.

```
echo "<pre>";
print_r($_SESSION);
print_r($_POST);
echo "</pre>";
```

11. How to implement the logout function?

In the above step, we added a "logout" button in main.php. When the logout button is pressed, the current session should be destroyed and login form should be displayed. Where should we put the code? The most appropriate place will be in login.php.

- Add the action to the form where logout button is defined.
- Then implement the action in login.php. You may use the code here as a reference:

```
if (isset($_POST["logout"])) {  
    session_destroy();  
}
```

12. (Optional in Fall 2023) Implement the submit function in quiz.html

In previous step, you had finished quiz.html as this. Now let's implement the submit function.

Q1: The pace of this course

- ☒ A: is too slow
- ☐ B: is just right
- ☐ C: is too fast
- ☐ D: I don't know

Q2: The feedback from homework assignment grading

- ☐ A: too few
- ☒ B: sufficient
- ☐ C: I don't know

Q3: Any thing you like about the teaching of this course?

Submit

12.1 Assign **name** attribute and **value** attribute for each input.

For the four radio input of Q1, set the name to be "Q1" and value to be "A","B","C","D"
For the three radio input of Q2, set the name to be "Q2" and value to be "A","B","C","D"
For Q3, set the name to be "Q3"

12.2 Add code to print user's input

- 1) Rename quiz.html to be quiz.php.
- 2) Set the form action to be quiz.php,
- 3) Add code to print the user's input. It should look like this:

Your answers are:

Q1:A

Q2:B

Q3:Labs

```
<?php
if (isset($_POST["submit"])) {
    echo "Your answers are: <br>";
    foreach (array_keys($_POST) as $x) {
        if ($x!='submit')
            echo $x . ":" . $_POST[$x] . "<br>";
    }
    return;
}
?>
```

13. Implement the submit function in question.html

In previous step, you had finished question.html as this.

Q1: The pace of this course

RemoveUpdate

Q2: The feedback from homework assignment grading

RemoveUpdate

Q3: Anything you like about the teaching of this course?

RemoveUpdate

Now let's implement the function for the buttons. When user clicked a button, you shall display message like this:

You clicked REMOVE for question 2

Or this:

You clicked UPDATE for question 3

Here is a simple to do:

- Use one form for each question.
- Assign **name** attribute and **value** attribute for each input.
- Use Hidden type to pass the question id.

```
<form method="post">
  Q1: The pace of this course
  <input type="hidden" name="q_id" value="1">
  <button type="submit" name="action" value="remove">Remove</button>
  <button type="submit" name="action" value="update">Update</button>
</form>
```

Add the following code to process the user action.

```
if (isset($_POST["action"])) {
    $action = $_POST["action"];
    switch ($action) {
        case "remove":
            echo " You clicked REMOVE for question " . $_POST['q_id'];
            break;
        case "update":
            echo " You clicked REMOVE for question " . $_POST['q_id'];
            break;
        default:
            echo "The action $action has no proessing code yet";
            break;
    }
} else {
    ?>
    #THIS is your html comment
    <?php
}
?>
```


Appendix: CSS

The picture below is a screen copy from Canvas.

Question 1	2 pts
Which linear data structure is good at random reading a data item?	
<input type="radio"/> Linked List	
<input type="radio"/> Array	

Question 2	2 pts
Which linear data structure is good at inserting or removing a time in the middle?	
<input type="radio"/> Linked List	
<input type="radio"/> Array	

Question 3	2 pts
Which abstract data structure has the property of First In First Out?	
<input type="radio"/> Stack	
<input type="radio"/> Queue	
<input type="radio"/> Priority Queue	

Submit Quiz

CSS

Topics	Example	Reference links
separate different areas.	<code><div></code> ... <code></div></code>	https://www.w3schools.com/tags/tag_div.asp
padding	padding: 20px The CSS Box Model: content → padding → border → Margin	https://www.w3schools.com/css/css_boxmodel.asp
border	border-top: solid border-color: lightgrey; border: 2px solid lightgrey	https://www.w3schools.com/css/css_border.asp

background-color	background-color: lightgrey	https://www.w3schools.com/cssref/pr_background-color.asp
float	float:left float:right clear:left floated content does not take up any space . Use overflow:auto	https://www.w3schools.com/css/css_float.asp
overflow		https://www.w3schools.com/css/css_overflow.asp
Special symbols	“Next” button is "Next ►"	